

Netskope, Inc. v. Fortinet, Inc.

IPR2023-00456

U.S. Patent No. 11,036,856

Patent Owner's Demonstratives

Hearing: May 31, 2024

Ground	Claim(s) challenged
1. Burdett, FortiSandbox	1-7, 9-18, 20-26, 28
2. Sharpe, Seese, Kamalapuram	1-2, 4-5, 7-10, 12-14, 16, 18-22, 24, 26-28
3. Sharpe, Seese, Kamalapuram, Dods	3, 6, 11, 15, 17, 23, 25

Topics of Discussion

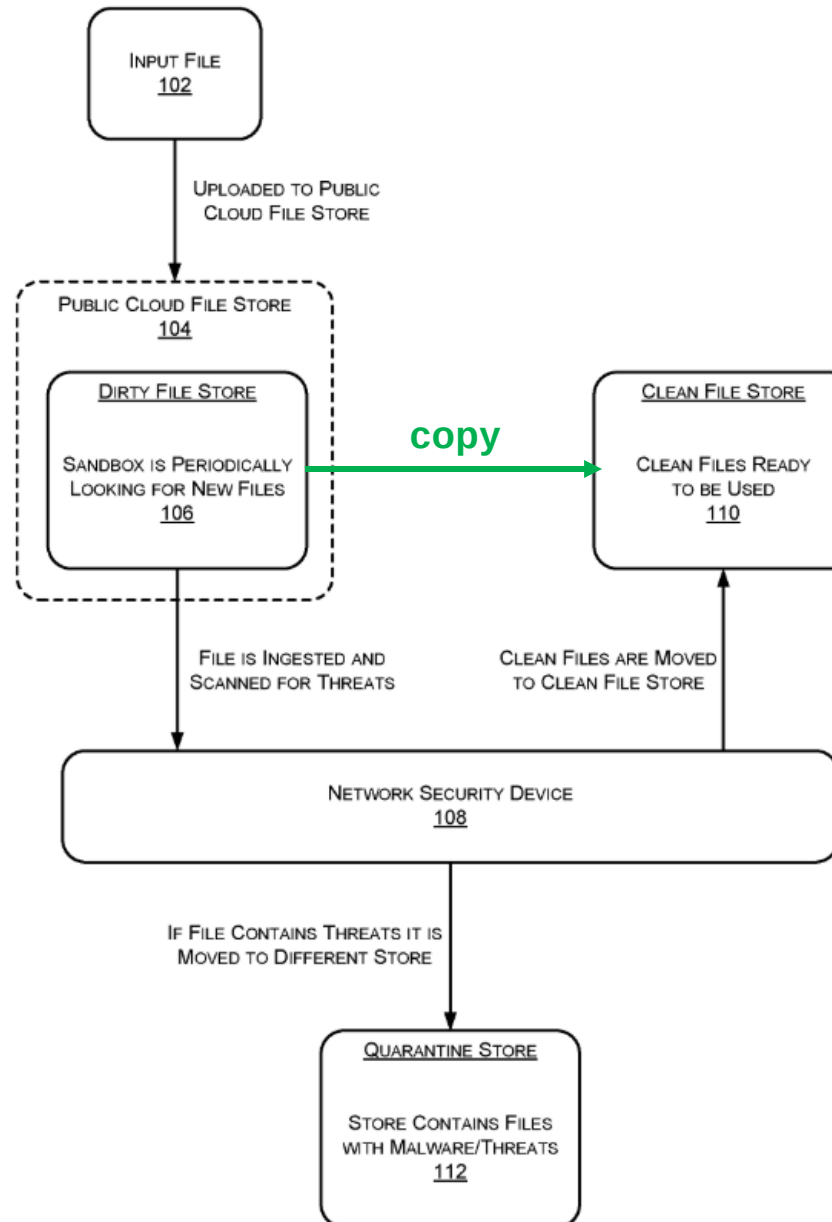
Ground	Patent Owner's Arguments	Applies to Claims
1 and 2	Petitioner's "moving" does not disclose or render obvious claimed "copying"	All Challenged Claims
1 and 2	Petitioner's proposed combinations introduce inefficiencies and are thus non-obvious	All Challenged Claims
2 (no corresponding theory in ground 1)	Petitioner has not shown it would be obvious to copy files from public cloud file store to outside of that public cloud file store	8, 19, and 27
3	No motivation to combine Sharpe, Seese, and Kamalapuram with Dods	3, 6, 11, 15, 17, 23, 25

**Petitioner's "Moving" of Files Does Not Render Obvious
Claimed "Copying"
(All Claims)**

'856 Patent Claims “Copying,” Not “Moving”

1. A secure data transfer system comprising:
 - a non-transitory storage device having embodied therein one or more routines operable to prevent users of an enterprise network from accessing malware infected files that are stored within public cloud file stores; and
 - one or more processors coupled to the non-transitory storage device and operable to execute the one or more routines, wherein the one or more routines include:
 - an untrusted file processing module, which when executed by the one or more processors:
 - accesses an untrusted file stored within a first repository of a public cloud file store, wherein the public cloud file store is natively mounted on a network security device that is protecting the enterprise network and wherein the users do not have read access to the first repository of the public cloud file store; and
 - causes the network security device to make a determination regarding whether the untrusted file is a clean file that is free of malicious content by applying one or more security checks to the untrusted file; and
 - a clean file transfer module, which when executed by the one or more processors, makes the clean file accessible to the users by, when a result of the determination is affirmative, copying the clean file from the first repository to a second repository that is accessible to the users.

'856 Patent Claims “Copying,” Not “Moving”



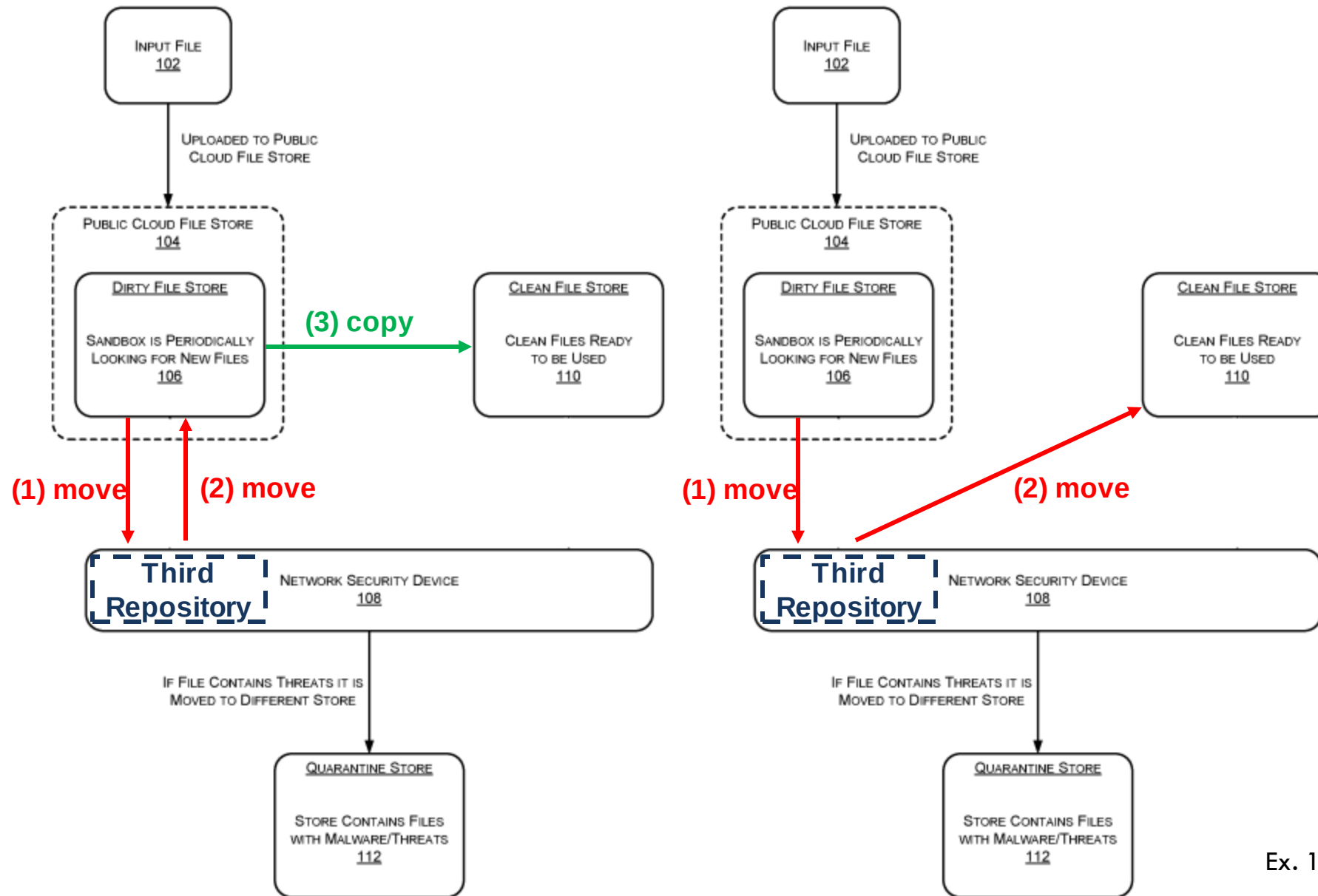
Ex. 1001 (856 patent) at FIG. 1B

'856 Patent Claims Distinguish “Copying” From “Moving”

1. A secure data transfer system comprising:
a non-transitory storage device having embodied therein one or more routines operable to prevent users of an enterprise network from accessing malware infected files that are stored within public cloud file stores; and one or more processors coupled to the non-transitory storage device and operable to execute the one or more routines, wherein the one or more routines include:
an untrusted file processing module, which when executed by the one or more processors:
accesses an untrusted file stored within a first repository of a public cloud file store, wherein the public cloud file store is natively mounted on a network security device that is protecting the enterprise network and wherein the users do not have read access to the first repository of the public cloud file store; and
causes the network security device to make a determination regarding whether the untrusted file is a clean file that is free of malicious content by applying one or more security checks to the untrusted file; and
a clean file transfer module, which when executed by the one or more processors, makes the clean file accessible to the users by, when a result of the determination is affirmative, copying the clean file from the first repository to a second repository that is accessible to the users.

12. The secure data transfer system of claim 1, wherein the network security device makes said determination regarding whether the untrusted file is a clean file that is free of malicious content by first moving the untrusted file from the first repository to a third repository, and then applying one or more security checks to the untrusted file in the third repository, and moving back the checked file as a clean file into the first repository for onward copying to the second repository or moving the checked file as a clean file directly from the third repository to the second repository.

'856 Patent Claims Distinguish “Copying” From “Moving”



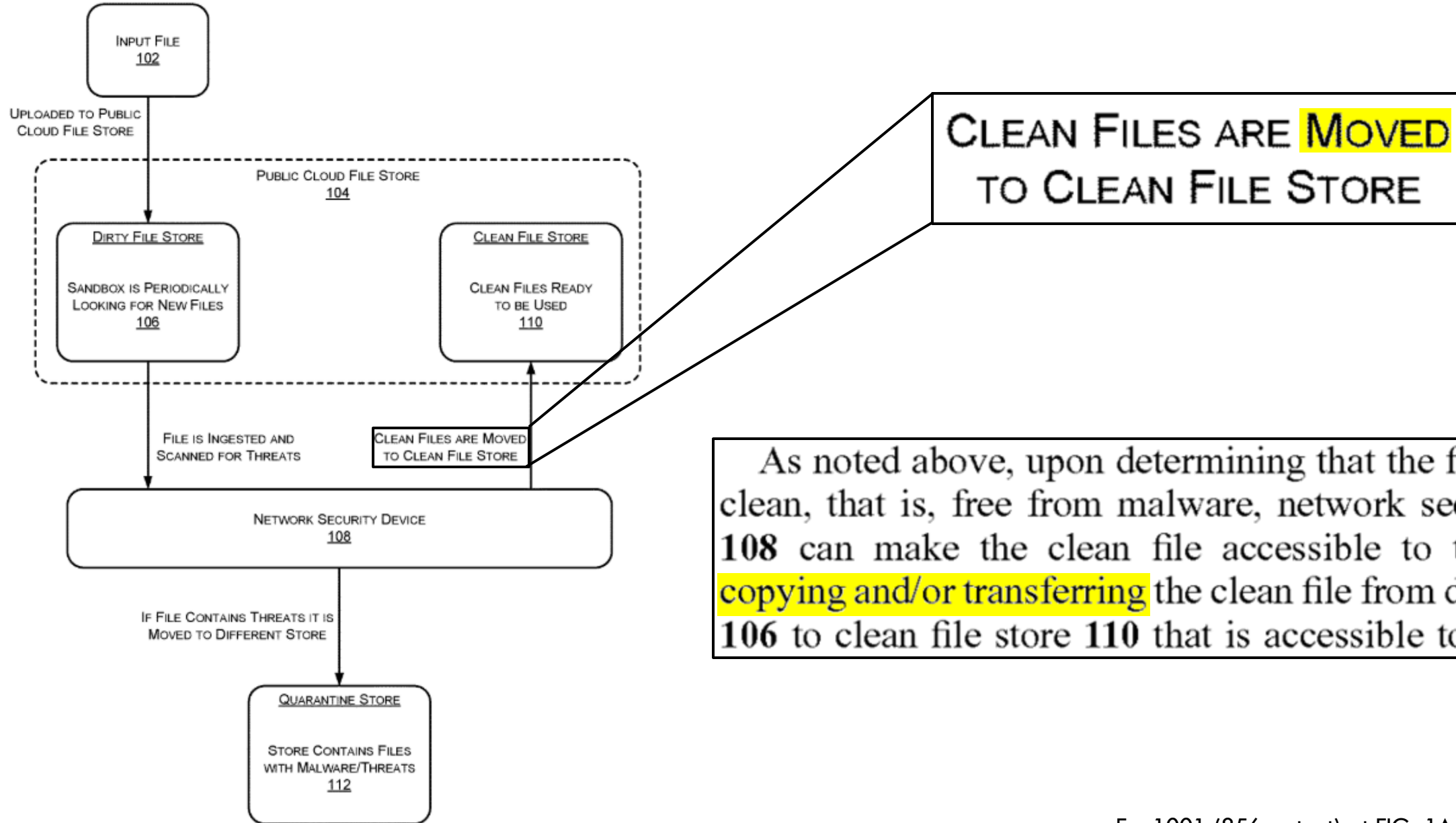
Ex. 1001 (856 patent) at Fig. 1B
(annotated); Sur-Reply at 3

“Copying” and “Moving” Are Not Synonymous

Q Is it your position that the claimed language "copying" is synonymous with "moving a file"?

A I never said that. I said that the patent specification uses these terms somewhat interchangeably. In particular, in Figures 1A and 1B, you know, it actually uses the term "moving" rather than "copying."

'856 Patent Purposely Claims "Copying"



As noted above, upon determining that the file at issue is clean, that is, free from malware, network security device 108 can make the clean file accessible to the users by copying and/or transferring the clean file from dirty file store 106 to clean file store 110 that is accessible to the users.

'856 Patent Purposely Claims "Copying"

Q. So if copying is the more inefficient operation, why does the '856 use it?

MR. ZIVOJNOVIC: Objection. Vague.

THE WITNESS: There could be an advantage even though, like we've described, it's more burdensome. For example, for a security reason, perhaps, you might want to take the files and physically isolate them from the suspect or potentially dirty files by moving them to a completely separate location so that there's no accidental potential intermixing.

This concept isn't described anywhere in the prior art, of course. This is something I'm reading from the '856.

You said far away in a separate location. What do you mean by that? You mean a separate physical media?

A. Perhaps. Perhaps in the separate S3 bucket, in a separate geographical location somehow isolated from where the dirty files or the potentially dangerous files live.

If you were to just update the metadata and just simply move from one location to another within the disk, you're not gaining that separation.

Petitioner's Discussion of "Copy-on-Write" Is Irrelevant

Q. So I believe you said that you're aware of at least a couple of counterexamples where copying doesn't necessarily include making a second copy of the file; is that right?

THE WITNESS: I'm aware of this other term. It's not called copying. It's called copy-on-write. Which could suggest, perhaps, to someone who's not familiar with the term, that it's making a separate copy. But instead, it's using a trick with links, and it's just making a link to the same copy. And if you read the documentation on copy-on-write, it will describe what it's actually doing, and say it's not doing a copy. It's actually just making a link to the same copy.

Petitioner's Discussion of "Copy-on-Write" Is Irrelevant

19. In such copy-on-write file systems, additional disk storage blocks are allocated only at the time that one of the copies is modified (i.e., "upon writing"), to store the modified bits of the affected copy—while any remaining copies continue to share the original bits.

Q Is it fair to say that you're relating the words "upon writing" to the words "on write"?

THE WITNESS: Yes. I'm giving an example, and I'm saying, see. Look, this is upon writing, and that sounds very similar to the known term "copy-on-write."

Petitioner's Discussion of "Copy-on-Write" Is Irrelevant

Copy-on-write method

In the copy-on-write method, when a change to the original volume occurs (but before the write I/O request is completed), each block to be modified is read and then written to the volume's shadow copy storage area (also called its "diff area"). The shadow copy storage area can be on the same volume or a different volume. This preserves a copy of the data block on the original volume before the change overwrites it.

Expand table

Time	Source data (status and data)	Shadow copy (status and data)
T0	Original data: 1 2 3 4 5	No copy: —
T1	Data changed in cache: 3 to 3'	Shadow copy created (differences only): 3
T2	Original data overwritten: 1 2 3' 4 5	Differences and index stored on shadow copy: 3

Q Sir, my question is very simple. Are you saying that this Microsoft article is misusing the word "copy-on-write"? Yes or no? That's all I'm asking.

THE WITNESS: Again, they have a very specific service, and they are using three catchy terms to describe the different methods, and copy-on-write is probably not an inappropriate term to relate to a person using this service of what it does.

Burdett Does Not Disclose Claimed "Copying"

Q Does Burdett disclose the claimed copying? Yes or no?

MS. SCHUNTER: Objection to form.

THE WITNESS: Burdett does not -- first of all, what is the claimed copying? That -- the patent uses these terms itself interchangeably, and Burdett teaches a mechanism that makes the clean file accessible only after the determination has been made that it's clean.

Q So does Burdett disclose the claimed copying as you interpret the term "copying" in the claim language?

MS. SCHUNTER: Objection. Form. Asked and answered.

THE WITNESS: Burdett teaches enough that a person of ordinary skill in the art would have found it obvious that one alternative to control access to a contaminated file was to move the file to a different location. And a person of ordinary skill would have understood that that is what this is all about; mainly, the end result that the clean file is accessible to the end user.

Burdett Does Not Disclose Claimed “Copying”

SUMMARY

It is desired to provide protection against compromise (e.g., malware or other exploits) or confidential information exfiltration in cloud service environments in a manner that is simple and efficient, and with minimal performance impact on applications using the cloud computing services. It is desirable to accomplish this without use of an agent installed on the template image used in cloud-based environments. This may be accomplished in some implementations with an architecture that connects a scanning service directly to the cloud data storage associated with a target application, in a manner that is intended to be efficiently configured and managed.

The scanning service 304 accesses files to be scanned directly from the cloud data store 306 that is used by the virtual machine instances, which avoids overhead and performance delay. The use of virtual machine instances for the scanning that are different from the virtual machine instances of the application facilitate management and reduce complexity. In some implementations, the scanning service provides alerts to an administrator, but does not attempt to control access to files. In some implementations, the scanning service may move files or change the name of files in order to control access. For example, to prevent access to a file, the scanning service may change the name or the location (e.g., path in a file system) of a file in order to prevent access. In some implementations, the scanning service 304 may replace a file with another file that is “clean.”

Burdett Does Not Render Obvious Claimed "Copying"

Q. How much faster is moving versus copying?

A. Well, here, I'm talking about that first type of move, where all you do is update the metadata and you're done. And I'm distinguishing it from the second type of move, which requires a copy. And I'm calling that copying data in the second part of the sentence.

So in the case where you actually have to -- to take the file contents and reproduce them, either you're doing a move type 2 or you're doing a copy, that is going to be slowed, especially depending on very large files. So if you're moving or copying a terabyte of data, that's going to take, you know, 20 minutes, an hour. I mean, it could be very long.

If you're simply moving that terabyte within the file system it already resides in, it's going to be instantaneous, virtually. It'll take a few milliseconds.

Burdett Does Not Render Obvious Claimed “Copying”

“Rejections on obviousness grounds,” in particular, “cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”

TQ Delta, LLC v. Cisco Systems, Inc., 942 F.3d 1352, 1359 (Fed. Cir. 2019)

The relevant inquiry is whether or not a person having ordinary skill in the art would be motivated to combine references despite drawbacks that the combination might present. ... The Board must weigh the benefits and drawbacks of the modification against each other, to determine whether there would be a motivation to combine.

Arctic Cat Inc. v. Polaris Industries, Inc., 795 F. App'x 827, 833 (Fed. Cir. 2019)

Burdett Does Not Render Obvious Claimed “Copying”

Guided by Burdett, a POSITA would have found it obvious that one alternative to control access to a contaminated file was to move the file to a different location—to a second repository - as opposed to changing the file permissions, and that this alternative would have been more efficient than managing access via file-level permissions. A POSITA would have found a further suggestion for this in Burdett’s Figure 3, which depicted storing clean (aqua in annotated Figure 3 below) and malicious (purple) files in separate locations. (Franz, ¶¶94-95.)

In light of Burdett’s disclosure, it would also have been obvious to copy a clean file to a second repository that is accessible to the applications. Additionally, by making a file accessible to one or more applications, a POSITA would have understood that the file would be available to users of those applications. (Franz, ¶96.)

Burdett Does Not Disclose Claimed “Copying”

Burdett’s disclosure is not limited to moving a file within the same file system – the “moving without copying” species described by PO. Burdett’s scanning service is configured to access data from multiple storage volumes. (Burdett 23:38 (“one or more cloud data stores 306”); 28:2-3 (“multiple data stores (e.g., multiple S3 buckets).”).) Since the scanning service can retrieve files from multiple volumes, a POSITA would have found it obvious to configure the scanning service to move files into multiple volumes after determining the files are clean, including moving a file from a dirty file repository on one volume to a clean file repository on another, different volume. In this case, Burdett’s “move” could not have been affected by merely changing a path in a file system. (Franz Reb., ¶¶23-26.) Indeed, Dr. Black

Burdett Does Not Render Obvious Claimed “Copying”

2. Burdett-FortiSandbox renders the “copying” limitation obvious.

Each of PO’s arguments against the obviousness of “copying” a file in view of Burdett’s disclosure fails.

- a. Burdett’s disclosure of “moving” renders the claimed “copying” obvious.**
- b. Burdett’s disclosure renders “copying” obvious even under PO’s erroneous construction.**
 - (i) Burdett discloses “moving,” a genus that includes “copying.”**
 - (ii) Burdett likewise discloses moving with copying.**
- c. A “copy” operation would not necessarily impact efficiency of Burdett’s system.**

3. Sharpe-Seese-Kamalapuram renders obvious the “copying” limitation.

PO’s arguments for nonobviousness of “copying” a file in view of Sharpe-Seese-Kamalapuram fail for similar reasons to those discussed for Burdett-FortiSandbox.

Burdett Does Not Render Obvious Claimed “Copying”

We agree with Atofina that the district court clearly erred in finding that JP 51–82206 anticipates the '514 patent. Anticipation requires a showing that each limitation of a claim is found in a single reference, either expressly or inherently. ... It is well established that the disclosure of a genus in the prior art is not necessarily a disclosure of every species that is a member of that genus.

Atofina v. Great Lakes Chem. Corp., 441 F.3d 991, 999 (Fed. Cir. 2006)

[O]ur predecessor court found a “pattern of preferences” in *Petering and Schaumann*. In this case, however, there is no such clear “pattern of preferences” that serves to narrow the genus in claim 2 to a narrow class that includes clopidogrel bisulfate. Even had claim 2 included salts generically, there was no expressed preference for clopidogrel bisulfate.

Sanofi-Synthelabo v. Apotex, Inc., 470 F.3d 1368, 1377-78 (Fed. Cir. 2008).

Burdett Does Not Render Obvious Claimed “Copying”

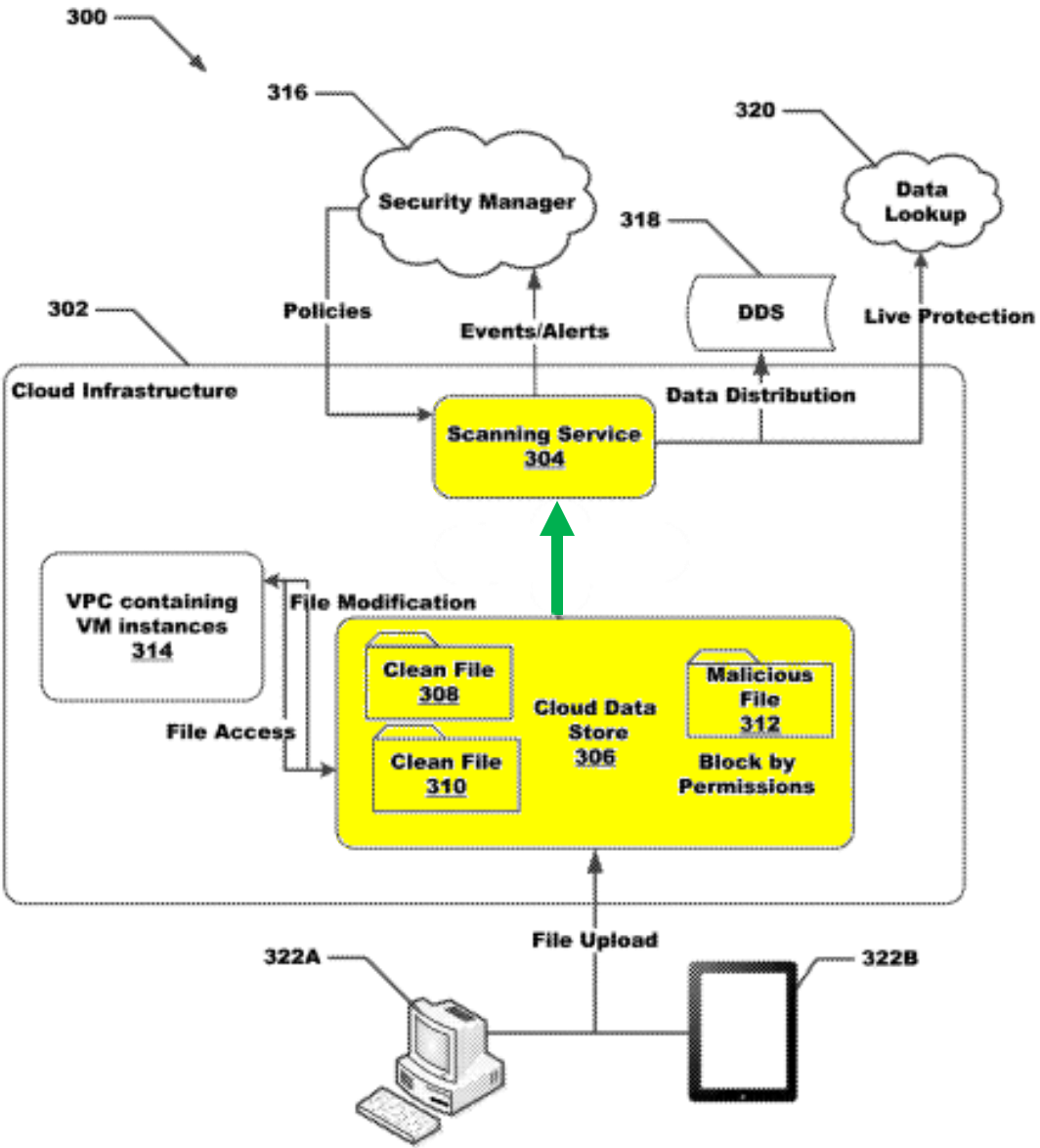
To prove obviousness under an obvious to try theory, Alkem must show (1) a design or market need to solve a particular problem, and (2) that “there are a finite number of identified, predictable solutions” that would lead to an expectation of success.

Grunenthal GMBH v. Alkem Labs. Ltd., 919 F.3d 1333, 1345 (Fed. Cir. 2019)

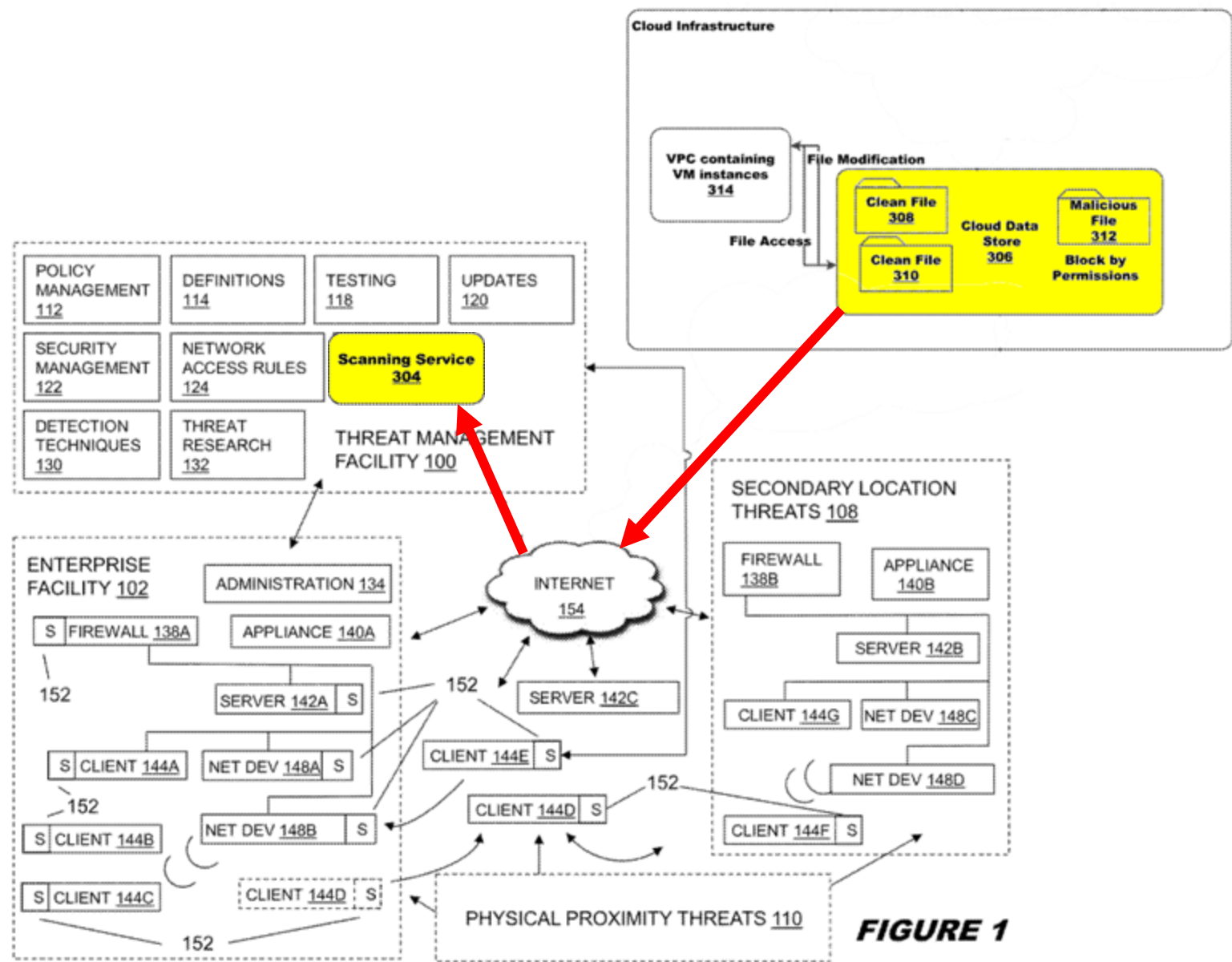
A particular course or selection is not obvious to try unless some design need or market pressure or other motivation would suggest to one of ordinary skill to pursue the claimed course or selection. ... In other words, one of ordinary skill must have “good reason to pursue the known options within his or her technical grasp.”

Rolls-Royce, PLC v. United Techs. Corp., 603 F.3d 1325, 1339 (Fed. Cir. 2010)

Petitioner's Combinations Are Not Obvious (All Claims)



Petitioner's Combination



Sharpe's Disclosure

FIG. 42 illustrates an exemplary scenario in which a cloud controller 4208 that is co-located with a cloud storage system 302 performs anti-virus scans for all of the data that is written to the distributed filesystem. More specifically, the cloud storage provider environment 4210 that hosts the cloud storage system 302 also hosts cloud compute services 4212 that include cloud controller 4208 (which executes as a virtual machine in cloud compute services 4212) and an anti-virus service 4214 that is executed in a virtual machine.

priately. This anti-virus scanning configuration offers a number of substantial performance benefits: (1) offloading anti-virus checks from cloud controllers that provide client file services reduces the load on these cloud controllers, thereby improving file performance for clients; (2) the virtual machine executing the cloud controller 4208 does not

require additional dedicated hardware and executes in the same data center as cloud storage system 302, and hence can communicate with cloud storage system 302 via a very low-latency, high-bandwidth connection to download the data blocks for new files very quickly; (3) the virtual machines executing cloud controller 4208 and anti-virus service 4214 may execute on the same server (and/or within the same data center), thereby also facilitating the rapid transfer of files and scan results between the two services; (4) the anti-virus scan executes completely “in the cloud” (e.g., within the cloud storage provider environment), and thus does not consume additional network resources for the other cloud controllers or client sites beyond the receipt of an additional incremental metadata snapshot indicating the results of the scan; and (5) network and compute bandwidth within the cloud are often cheaper than network bandwidth from a client site to the cloud storage system and compute cycles in physical cloud controllers—thus, an arrangement that reduces the load on an organization’s wide-area network links and physical cloud controllers can reduce cost while increasing performance. Note, however, that executing a

No Motivation to Combine

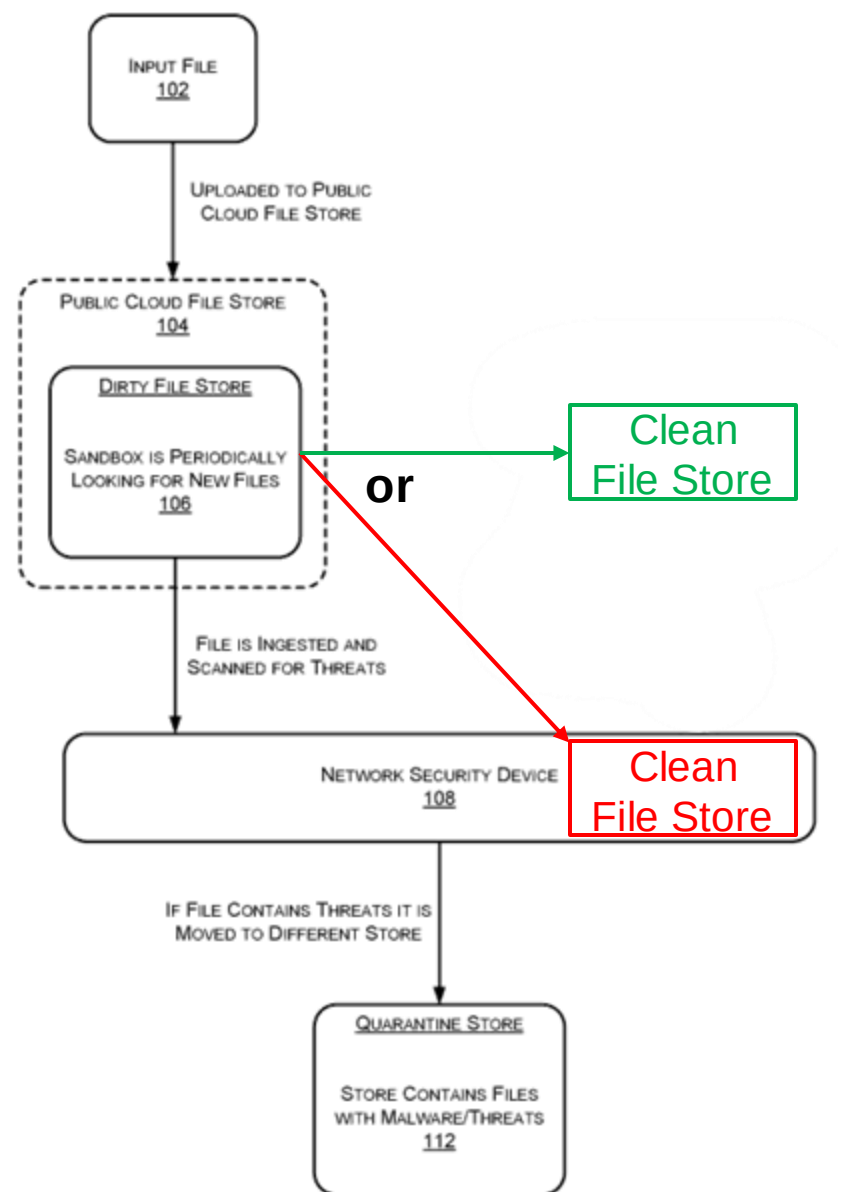
First, PO's alleged drawbacks are vague, conclusory, and hypothetical. (*See* Supp. Black Dec., ¶ 65 (combination “*may* result in network congestion” and “there is always *the possibility* of a third party snooping traffic”).) Importantly, PO provides no credible evidence that these alleged drawbacks outweigh benefits of the proposed combination, which provides improved security and efficiency. (Franz Reb., ¶¶37-38); *see, e.g., Intel Corp. v. PACT XPP Schweiz AG*, No. 2022-1038,

4214”).) Thus, for similar reasons as those discussed above with respect to Burdett, a POSITA would not have found it prohibitively inefficient to perform Sharpe's anti-virus scan on a network security device such as Seese's because Sharpe already contemplates files or portions of files being transferred from cloud storage to an anti-virus service. (Franz Reb., ¶¶40-41.)

increasing performance. Note, however, that executing a cloud controller in a cloud storage provider environment implies trust in this environment; the cloud controller needs to decrypt the files before transferring them to the anti-virus service, and hence if the environment is compromised the files may be readable by other parties. If this risk is unacceptable, another substantially similar configuration could use a dedicated physical cloud controller and an anti-virus service that are located at a secure site in close network proximity with the cloud storage system (and have a high-bandwidth link to the cloud storage system) to perform anti-virus scans for the distributed filesystem.

**Not Obvious to Copy Files Outside of Public Cloud File Store
(Dependent Claims 8, 19, and 27)**

Dependent Claims 8, 19, and 27



8. The secure data transfer system of claim 1, wherein the second repository is part of the network security device or part of an external storage device that forms part of the enterprise network.

Seese' Disclosure of Different Cloud Stores

